IN THE CLAIMS

Claims 1-22 are presented below, with claims 1-2 and 4-22 pending. As shown below, claims 1, 15, and 18 have been amended.

1. (Currently Amended) Method for communication between an application program and a network device driver program through intermediate structure software, comprising the steps of:

a. supplying of application data units from the application program to a first program object being part of the intermediate structure software;

b. performing of first functions of the first program object on the application data units;

c. supplying of resulting first data units from the first program object to a second program object being part of the intermediate structure software;

d. performing of second functions of the second program object on the first data units;

e. supplying of the resulting second data units to the network device driver program;

wherein supplying data units between program objects is accomplished by passing references pointing to memory locations storing data of the data units such that the references are passed between program objects and the data of the data units is not passed directly between program objects, and

wherein for at least one application data unit, the referenced memory location storing data of the application data unit is the same memory location as the referenced memory location storing at least some of the data of the corresponding first data unit and as the referenced memory location for storing at least some of the data of the corresponding second data unit.

2. (Original) Method according to claim 1, wherein data units are supplied over interconnecting queue-objects.

3. (Canceled)

4. (Original) Method according to claim 2, wherein data units are supplied over interconnecting queue-objects, wherein the queue-objects have different priorities.

5. (Previously Presented) Method according to claim 1, wherein program objects are added during run time of the application program.

6. (Previously Presented) Method according to claim 1, wherein program objects are removed during run time of the application program.

7. (Previously Presented) Method according to claim 1, wherein after performing of functions of a program object and supplying the data units to a further program object additional functions of the program object are performed.

8. (Previously Presented) Method according to claim 2, wherein step a and/or c also comprises adding or removing information to or from said data units.

9. (Previously Presented) Method according to claim 1, also comprising dividing data units into data units parts or uniting data unit parts into data units.

10. (Previously Presented) Method according to claim 1, providing service data units containing one or more data units.

11. (Original) Method according to claim 10, referencing data units with a reference to the service data unit.

12. (Previously Presented) Method according to claim 1, also providing a specialized execution environment for communication between the application program and the network device driver program.

13. (Previously Presented) Method according to claim 1, wherein data units are organized in data unit pools adapted to the specific use thereof.

14. (Previously Presented) Method according to claim 1, providing a naming service for mapping between the internal communication mechanism of the hardware and symbolic names.

15. (Currently Amended) System for communication between an application program and a network device driver program and vice versa through intermediate structure software, comprising.

a. a first program object being part of the intermediate structure software and for performing of first functions on data units, said data units being transferred to and from the application program and data units being transferred to and from said first program object;

b. a second program object being part of the intermediate structure software and for performing of second functions on said data units, said data units being transferred to and from said second program object and data units being transferred to and from the network driver;

wherein transferring data units between program objects is accomplished by passing references pointing to memory locations storing data of the data units such that the references are passed between program objects and the data of the data units is not passed directly between program objects, and

wherein for at least one data unit, data of the data unit is not moved from the referenced memory location of that data unit to a different memory location while the first program object performs said first functions and while the second program object performs said second functions.

16. (Original) System according to claim 15, wherein service data units are stored in a memory part using references.

17. (Previously Presented) System according to claim 15, provided with a SDU manager.

18. (Currently Amended) Method for communication between a network device driver program and an application program through intermediate structure software, comprising the steps of:

a. supplying of first data units form the network device driver program to a first program object or protocol object being part of the intermediate structure software;

b. performing of first functions of the first program object on said first data units;

c. supply of resulting second data units from the first program object to a second program object being part of the intermediate structure software;

d. performing of second functions of the second program object on the second data units;

e. supplying of resulting application data units from the second program object to said application program;

wherein supplying data units between program objects is accomplished by passing references pointing to memory locations storing data of the data units such that the references are passed between program objects and the data of the data units is not passed directly between program objects, and

wherein for at least one application data unit, the referenced memory location storing data of the application data unit is the same memory location as the referenced memory location storing at least some of the data of the corresponding first data unit and as the referenced memory location for storing at least some of the data of the corresponding second data unit.

19. (Original) Method according to claim 4, wherein within a queue-object two or more priorities for passing of data units are provided.

20. (Previously Presented) Method according to claim 10, wherein at least two data units referenced by a service data unit are stored in non-contiguous portions of memory.

21. (Previously Presented) Method according to claim 12, wherein the specialized execution environment forms a plurality of network protocol layers and the first program object and the second program object are in respective network protocol layers.

22. (Previously Presented) Method according to claim 1, further comprising creating a service data unit for each application data unit, each service data unit including a size value indicating the size of data of the application data unit and an offset value indicating the memory location storing data of the application data unit,

wherein supplying data units between program objects by passing references includes passing service data units corresponding to the supplied data units.